PCT ORGANISATION MONDIALE DE LA PROPRIETE INTELLECTUELLE
Bureau international

DEMANDE INTERNATIONALE PUBLIEE EN VERTU DU TRAITE DE COOPERATION EN MATIERE DE BREVETS (PCT)

(51) Classification internationale des brevets 7 :		(11) Numéro de publication internationale:
H04L 9/06	A1	WO 00/27068
		(43) Date de publication internationale: 11 mai 2000 (11.05.00)

(21) Numéro de la demande internationale: PCT/FR99/02660

(22) Date de dépôt international: 29 octobre 1999 (29.10.99)

(30) Données relatives à la priorité:
98/13605 29 octobre 1998 (29.10.98) FR

(71) Déposant *(pour tous les Etats désignés sauf US)*: GEMPLUS S.C.A. [FR/FR]; Avenue du Pic de Bertagne, Parc d'Activités de Gémenos, F–13881 Gémenos Cedex (FR).

(71)(72) Déposants et inventeurs: CLAVIER, Christophe [FR/FR]; 5, rue de la République, F–13420 Gémenos (FR). CORON, Jean–Sébastien [FR/FR]; 4, rue De Lagrange, F–75015 Paris (FR).

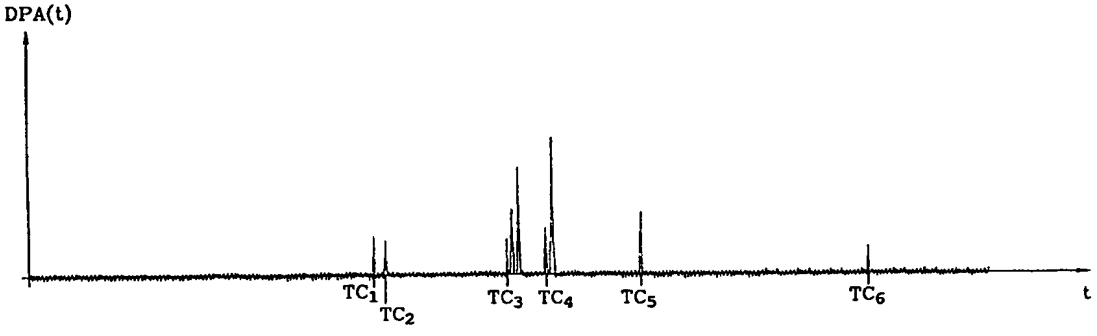
(74) Mandataire: NONNENMACHER, Bernard; Gemplus S.C.A., Avenue du Pic de Bertagne, Parc d'Activités de Gémenos, F–13881 Gémenos Cedex (FR).

(81) Etats désignés: AU, BR, CA, CN, IN, JP, KR, MX, RU, SG, US, VN, brevet européen (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).

Publiée
Avec rapport de recherche internationale.

(54) Title: COUNTERMEASURE METHOD IN AN ELECTRONIC COMPONENT USING A SECRET KEY CRYPTOGRAPHIC ALGORITHM

(54) Titre: PROCEDE DE CONTRE–MESURE DANS UN COMPOSANT ELECTRONIQUE METTANT EN OEUVRE UN ALGORITHME DE CRYPTOGRAPHIE A CLE SECRETE



(57) Abstract

In an electronic component using a secret key K cryptographic algorithm, the algorithm operation comprises the use of first means TC_0 for supplying output data S from an input data E, the output data and/or derived data being manipulated by critical instructions, a countermeasure method consists in using other means TC_1, TC_2 such that the output data and the derived data are unpredictable, said other means being obtained from said first means by an or else operation with a random value u or a derived random value e(p(u) on one and/or the other of the input and output data of said first means TC_0.

(57) **Abrégé**

Dans un composant électronique mettant en oeuvre un algorithme cryptographique à clé secrète (K), la mise en oeuvre de l'algorithme comprenant l'utilisation de premiers moyens TC_0 pour fournir une donnée de sortie (S) à partir d'une donnée d'entrée (E), la donnée de sortie et/ou des données dérivées étant manipulées par des instructions critiques, un procédé de contre–mesure prévoit l'utilisation d'autres moyens TC_1, TC_2, en sorte que la donnée de sortie et les données dérivées soient imprédictibles, ces autres moyens étant obtenus des dits premiers moyens par une opération de OU EXCLUSIF avec une valeur aléatoire (u) ou une valeur aléatoire dérivée e(p(u) sur l'une et/ou sur l'autre des données d'entrée et de sortie des dits premiers moyens TC_0.

UNIQUEMENT A TITRE D'INFORMATION

Codes utilisés pour identifier les Etats parties au PCT, sur les pages de couverture des brochures publiant des demandes internationales en vertu du PCT.

AL	Albanie	**ES**	Espagne	**LS**	Lesotho	**SI**	Slovénie
AM	Arménie	**FI**	Finlande	**LT**	Lituanie	**SK**	Slovaquie
AT	Autriche	**FR**	France	**LU**	Luxembourg	**SN**	Sénégal
AU	Australie	**GA**	Gabon	**LV**	Lettonie	**SZ**	Swaziland
AZ	Azerbaïdjan	**GB**	Royaume-Uni	**MC**	Monaco	**TD**	Tchad
BA	Bosnie-Herzégovine	**GE**	Géorgie	**MD**	République de Moldova	**TG**	Togo
BB	Barbade	**GH**	Ghana	**MG**	Madagascar	**TJ**	Tadjikistan
BE	Belgique	**GN**	Guinée	**MK**	Ex-République yougoslave de Macédoine	**TM**	Turkménistan
BF	Burkina Faso	**GR**	Grèce			**TR**	Turquie
BG	Bulgarie	**HU**	Hongrie	**ML**	Mali	**TT**	Trinité-et-Tobago
BJ	Bénin	**IE**	Irlande	**MN**	Mongolie	**UA**	Ukraine
BR	Brésil	**IL**	Israël	**MR**	Mauritanie	**UG**	Ouganda
BY	Bélarus	**IS**	Islande	**MW**	Malawi	**US**	Etats-Unis d'Amérique
CA	Canada	**IT**	Italie	**MX**	Mexique	**UZ**	Ouzbékistan
CF	République centrafricaine	**JP**	Japon	**NE**	Niger	**VN**	Viet Nam
CG	Congo	**KE**	Kenya	**NL**	Pays-Bas	**YU**	Yougoslavie
CH	Suisse	**KG**	Kirghizistan	**NO**	Norvège	**ZW**	Zimbabwe
CI	Côte d'Ivoire	**KP**	République populaire démocratique de Corée	**NZ**	Nouvelle-Zélande		
CM	Cameroun			**PL**	Pologne		
CN	Chine	**KR**	République de Corée	**PT**	Portugal		
CU	Cuba	**KZ**	Kazakstan	**RO**	Roumanie		
CZ	République tchèque	**LC**	Sainte-Lucie	**RU**	Fédération de Russie		
DE	Allemagne	**LI**	Liechtenstein	**SD**	Soudan		
DK	Danemark	**LK**	Sri Lanka	**SE**	Suède		
EE	Estonie	**LR**	Libéria	**SG**	Singapour		

PROCÉDÉ DE CONTRE-MESURE DANS UN COMPOSANT ÉLECTRONIQUE METTANT EN OEUVRE UN ALGORITHME DE CRYPTOGRAPHIE A CLÉ SECRETE

La présente invention concerne un procédé de contre-mesure dans un composant électronique mettant en oeuvre un algorithme de cryptographie à clé secrète. Ils sont utilisés dans des applications où l'accès à des services ou à des données est sévèrement contrôlé. Ils ont une architecture formée autour d'un microprocesseur et de mémoires, dont une mémoire programme qui contient la clé secrète.

Ces composants sont notamment utilisés dans les cartes à puce, pour certaines applications de celles-ci. Ce sont par exemple des applications d'accès à certaines banques de données, des applications bancaires, des applications de télépéage, par exemple pour la télévision, la distribution d'essence ou encore le passage de péages d'autoroutes.

Ces composants ou ces cartes mettent donc en oeuvre un algorithme de cryptographie à clé secrète, dont le plus connu est l'algorithme DES (pour *Data Encryption Standard* dans la littérature anglo-saxonne). D'autres algorithmes à clé secrète existent, comme l'algorithme RC5 ou encore l'algorithme COMP128. Cette liste n'est bien sûr pas exhaustive.

De manière générale et succincte, ces algorithmes ont pour fonction de calculer un message chiffré à partir d'un message appliqué en entrée (à la carte) par un système hôte (serveur, distributeur bancaire...) et de la clé secrète contenue dans la carte, et de fournir en retour au système hôte ce message chiffré, ce qui permet par exemple au système hôte d'authentifier le composant ou la carte, d'échanger des données...

Or il est apparu que ces composants ou ces cartes sont vulnérables à des attaques consistant en une analyse différentielle de consommation en courant et qui permettent à des tiers mal intentionnés de trouver la clé secrète. Ces attaques sont appelées attaques DPA, acronyme anglo-saxon pour *Differential Power Analysis*.

Le principe de ces attaques DPA repose sur le fait que la consommation en courant du microprocesseur exécutant des instructions varie selon la donnée manipulée.

Notamment, une instruction du microprocesseur manipulant un bit de donnée génère deux profils de courant différents selon que ce bit vaut "1" ou "0". Typiquement, si l'instruction manipule un "0", on a à cet instant d'exécution une première amplitude du courant consommé et si l'instruction manipule un "1", on a une deuxième amplitude du courant consommé, différente de la première.

Les caractéristiques des algorithmes de cryptographie sont connues : calculs effectués, paramètres utilisés. La seule inconnue est la clé secrète contenue en mémoire programme. Celle-ci ne peut être déduite de la seule connaissance du message appliqué en entrée et du message chiffré fourni en retour.

Cependant, dans un algorithme de cryptographie, certaines données calculées dépendent seulement du message appliqué en clair en entrée de la carte et de la clé secrète contenue dans la carte. D'autres données calculées dans l'algorithme peuvent aussi être recalculées seulement à partir du message chiffré (généralement fourni en clair en sortie de la carte vers le système hôte) et de la clé secrète contenue dans la carte. Plus précisément, chaque bit de ces données particulières peut être déterminé à partir du

message d'entrée ou de sortie, et d'un nombre limité de bits particuliers de la clé.

Ainsi, à chaque bit d'une donnée particulière, correspond une sous-clé formée par un groupe particulier de bits de la clé.

Les bits de ces données particulières qui peuvent être prédites sont appelés dans la suite, bits cibles.

L'idée de base de l'attaque DPA est ainsi d'utiliser la différence du profil de consommation en courant d'une instruction selon qu'elle manipule un "1" ou un "0" et la possibilité de calculer un bit cible par les instructions de l'algorithme à partir d'un message connu d'entrée ou de sortie et d'une hypothèse sur la sous-clé correspondante.

Le principe de l'attaque DPA est donc de tester une hypothèse de sous-clé donnée, en appliquant sur un grand nombre de courbes de mesure en courant, chacune relative à un message d'entrée connu de l'attaquant, une fonction booléenne de sélection, fonction de l'hypothèse de sous-clé, et définie pour chaque courbe par la valeur prédite pour un bit cible.

En faisant une hypothèse sur la sous-clé concernée, on est en effet capable de prédire la valeur "0" ou "1" que va prendre ce bit cible pour un message d'entrée ou de sortie donné.

On peut alors appliquer comme fonction booléenne de sélection, la valeur prédite "0" ou "1" par le bit cible pour l'hypothèse de sous-clé considérée, pour trier ces courbes en deux paquets : un premier paquet regroupe les courbes qui ont vu la manipulation du bit cible à "0" et un deuxième paquet regroupe les courbes qui ont vu la manipulation du bit cible à "1" selon l'hypothèse de sous-clé. En faisant la moyenne de consommation en courant dans chaque paquet, on obtient une courbe de consommation moyenne M0(t) pour le

premier paquet et une courbe de consommation moyenne M1(t) pour le deuxième paquet.

Si l'hypothèse de sous-clé est juste, le premier paquet regroupe réellement toutes les courbes parmi les N courbes qui ont vu la manipulation du bit cible à "0" et le deuxième paquet regroupe réellement toutes les courbes parmi les N courbes qui ont vu la manipulation du bit cible à "1". La courbe moyenne de consommation M0(t) du premier paquet aura alors une consommation moyenne partout sauf aux moments de l'exécution des instructions critiques, avec un profil de consommation en courant caractéristique de la manipulation du bit cible à "0" ($profil_0$). En d'autres termes, pour toutes ces courbes tous les bits manipulés ont eu autant de chances de valoir "0" que de valoir "1", sauf le bit cible qui a toujours eu la valeur "0". Ce qui peut s'écrire :

$M0(t)=[(profil_0+profil_1)/2]_{t\neq tci} + [profil_0]_{tci}$ soit

$M0(t)=[Vm_t]_{t\neq tci} + [profil_0]_{tci}$

où tci représente les instants critiques, auxquels une instruction critique a été exécutée.

De même, la courbe moyenne de consommation M1(t) du deuxième paquet correspond à une consommation moyenne partout sauf aux moments de l'exécution des instructions critiques, avec un profil de consommation en courant caractéristique de la manipulation du bit cible à "1" ($profil_1$). On peut écrire :

$M1(t)=[(profil_0+profil_1)/2]_{t\neq tci} + [profil_1]_{tci}$ soit

$M1(t)=[Vm_t]_{t\neq tci} + [profil_1]_{tci}$

On a vu que les deux profils $profil_0$ et $profil_1$ ne sont pas égaux. La différence des courbes M0(t) et M1(t) donne alors un signal DPA(t) dont l'amplitude est égale à $profil_0$-$profil_1$ aux instants critiques tci d'exécution des instructions critiques manipulant ce bit, c'est à dire, dans l'exemple représenté sur la figure 1, aux endroits tc0 à tc6 et dont l'amplitude

est à peu près égale à zéro en dehors des instants critiques.

Si l'hypothèse de sous-clé est fausse, le tri ne correspond pas à la réalité. Statistiquement, il y a alors dans chaque paquet, autant de courbes ayant vu réellement la manipulation du bit cible à "0" que de courbes ayant vu la manipulation du bit cible à "1". La courbe moyenne résultante M0(t) se situe alors autour d'une valeur moyenne donnée par $(profil_0+profil_1)/2=Vm$, car pour chacune des courbes, tous les bits manipulés, y compris le bit cible ont autant de chances de valoir "0" que de valoir "1".

Le même raisonnement sur le deuxième paquet conduit à une courbe moyenne de consommation en courant M1(t) dont l'amplitude se situe autour d'une valeur moyenne donnée par $(profil_0+profil_1)/2=Vm$.

Le signal DPA(t) fourni par la différence M0(t)-M1(t) est dans ce cas sensiblement égal à zéro. Le signal DPA(t) dans le cas d'une hypothèse de sous-clé fausse est représenté sur la figure 2.

Ainsi l'attaque DPA exploite la différence du profil de consommation en courant pendant l'exécution d'une instruction suivant la valeur du bit manipulé, pour effectuer un tri de courbes de consommation en courant selon une fonction de sélection booléenne pour une hypothèse de sous-clé donnée. En effectuant une analyse différentielle de la consommation moyenne en courant entre les deux paquets de courbes obtenus, on obtient un signal d'information DPA(t).

Le déroulement d'une attaque DPA consiste alors globalement:

a- à tirer N messages aléatoires (par exemple N égal 1000);

b- à faire exécuter l'algorithme par la carte pour chacun des N messages aléatoires, en relevant la courbe

de consommation en courant à chaque fois (mesurée sur la borne d'alimentation du composant);

c- à faire une hypothèse sur une sous-clé;

d- à prédire, pour chacun des messages aléatoires, la valeur prise par un des bits cibles dont la valeur ne dépend que des bits du message (d'entrée ou de sortie) et de la sous-clé prise en hypothèse, pour obtenir la fonction de sélection booléenne;

e- à trier les courbes selon cette fonction de sélection booléenne (c'est à dire selon la valeur "0" ou "1" prédite pour ce bit cible pour chaque courbe sous l'hypothèse de sous-clé);

f- à calculer dans chaque paquet la courbe résultante de consommation moyenne en courant;

g- à effectuer la différence de ces courbes moyennes, pour obtenir le signal DPA(t).

Si l'hypothèse sur la sous-clé est juste, la fonction de sélection booléenne est juste et les courbes du premier paquet correspondent réellement aux courbes pour lesquelles le message appliqué en entrée ou en sortie a donné un bit cible à "0" dans la carte et les courbes du deuxième paquet correspondent réellement aux courbes pour lesquelles le message appliqué en entrée ou en sortie a donné un bit cible à "1" dans la carte.

On est dans le cas de la figure 1 : le signal DPA(t) n'est donc pas nul aux instants tc0 à tc6 correspondant à l'exécution des instructions critiques (celles qui manipulent le bit cible).

On notera que l'attaquant n'a pas besoin de connaître avec précision les instants critiques. Il suffit qu'il y ait au moins un instant critique dans la période d'acquisition.

Si l'hypothèse de sous-clé n'est pas juste, le tri ne correspond pas à la réalité et on a alors dans chaque paquet autant de courbes correspondant en

réalité à un bit cible à "0" que de courbes correspondant à un bit cible à "1". Le signal DPA(t) est sensiblement nul partout (cas représenté à la figure 2). Il faut retourner à l'étape c- et faire une nouvelle hypothèse sur la sous-clé.

Si l'hypothèse s'avère juste, on peut passer à l'évaluation d'autres sous-clés, jusqu'à avoir reconstitué la clé au maximum. Par exemple, avec un algorithme DES, on utilise une clé de 64 bits, dont seulement 56 bits utiles. Avec une attaque DPA, on est capable de reconstituer au moins 48 bits des 56 bits utiles.

La présente invention a pour but de mettre en oeuvre dans un composant électronique, un procédé de contre-mesure qui entraîne un signal DPA(t) nul, même dans le cas où l'hypothèse de sous-clé est juste.

De cette façon, rien ne permet de distinguer le cas de l'hypothèse de sous-clé juste des cas d'hypothèses de sous-clé fausses. Par cette contre-mesure, le composant électronique est paré contre les attaques DPA.

Mais dans l'invention, on s'est rendu compte qu'il ne suffisait pas de faire en sorte que le signal DPA(t) soit nul relativement à un bit cible donné.

En effet, si on considère la valeur prise par plusieurs bits cibles d'une même donnée manipulée par les instructions critiques, on va devoir trier les courbes non plus en deux paquets, mais en plusieurs paquets. On n'a plus une fonction de sélection binaire. On peut montrer qu'en regroupant ensuite ces paquets d'une manière ou d'une autre, on peut obtenir un signal DPA(t) non nul dans le cas d'une hypothèse de sous-clé juste, alors qu'il aurait été nul si l'on aviat trié selon une fonction de sélection binaire sur un seul bit cible.

Prenons par exemple deux bits cibles d'une même donnée. Ces deux bits cibles peuvent prendre les 2^2 valeurs suivantes : "00", "01", "10" et "11".

En appliquant la fonction de sélection aux N=1000 courbes de consommation en courant mesurées, on obtient quatre paquets de courbes. Si le tri est juste, un premier paquet de 250 courbes environ correspond à la valeur "00", un deuxième paquet de 250 courbes environ correspond à la valeur "01", un troisième paquet de 250 courbes environ correspond à la valeur "10" et un quatrième paquet de 250 courbes environ correspond à la valeur "11".

Si on regroupe les premier et quatrième paquets dans un premier groupe et les deuxième et troisième paquets dans un deuxième groupe, on obtient deux groupes qui ne sont pas équivalents.

Dans le premier groupe, les deux bits ont autant de chances de valoir "00" que de valoir "11". La valeur moyenne aux instants critiques de toutes les courbes de consommation de ce groupe peut s'écrire :

M1 (t_{ci})=[consommation("00") + consommation ("11")]/2

Dans le deuxième groupe, les deux bits ont autant de chances de valoir "01" que de valoir "10". La valeur moyenne aux instants critiques de toutes les courbes de consommation de ce groupe peut s'écrire :

M2 (t_{ci})=[consommation("01") + consommation ("10")]/2

Si on fait la différence entre ces deux moyennes, on obtient un signal DPA(t) non nul. En d'autres termes, les deux groupes dont on compare les consommations moyennes n'ont pas un contenu équivalent.

Dans l'invention, on a donc cherché à empêcher l'obtention d'un quelconque signal significatif au sens de l'attaque DPA. Quel que soit le nombre de bits cibles pris, quelle que soit la combinaison de paquets effectuée pour faire la comparaison des consommations moyennes, le signal DPA(t) sera toujours nul. Il faut

donc obtenir des paquets équivalents, quel que soit le nombre de bits cibles considérés.

Une solution à ces différents problèmes techniques a été trouvée dans l'utilisation d'une valeur aléatoire dans une opération de OU EXCLUSIF avec l'une et/ou l'autre des données d'entrée et de sortie de moyens utilisés dans l'algorithme.

Avec une utilisation selon l'invention d'une telle valeur aléatoire, les données manipulées par les instructions critiques deviennent imprédictibles tout en ayant un résultat juste en sortie de l'algorithme.

Telle que caractérisée, l'invention concerne donc un procédé de contre-mesure dans un composant électronique mettant en oeuvre un algorithme cryptographique à clé secrète, la mise en oeuvre de l'algorithme comprenant l'utilisation de premiers moyens pour fournir une donnée de sortie à partir d'une donnée d'entrée, la donnée de sortie et/ou des données dérivées étant manipulées par des instructions critiques. Selon l'invention, le procédé de contre-mesure prévoit l'utilisation d'autres moyens, en sorte que la donnée de sortie et les données dérivées soient imprédictibles, ces autres moyens étant obtenus des dits premier moyens par une opération de OU EXCLUSIF avec une valeur aléatoire ou une valeur aléatoire dérivée sur l'une et/ou sur l'autre des données d'entrée et de sortie des dits premiers moyens.

D'autres caractéristiques et avantages de l'invention sont détaillés dans la description suivante faite à titre indicatif et nullement limitatif et en référence aux dessins annexés, dans lesquels :

- les figures 1 et 2 déjà décrites représentent le signal DPA(t) que l'on peut obtenir en fonction d'une hypothèse sur une sous-clé de la clé secrète K, selon une attaque DPA;

- les figures 3 et 4 sont des organigrammes détaillés des premiers et derniers tours de l'algorithme DES;

- la figure 5 est un schéma-bloc de l'opération SBOX utilisée dans l'algorithme DES;

- la figure 6 montre un exemple de table de constantes élémentaire à une entrée et une sortie utilisée dans l'opération SBOX;

- la figure 7 représente un premier exemple d'organigramme d'exécution du DES avec un procédé de contre-mesure selon l'invention;

- la figure 8 est un organigramme des premiers tours du DES correspondant;

- les figures 9 et 10 représentent respectivement un organigramme d'exécution du DES et un organigramme détaillé des premiers tours, dans un deuxième mode d'application du procédé de contre mesure selon l'invention;

- les figures 11 et 12 correspondent à un troisième mode d'application du procédé de contre mesure selon l'invention;

- la figure 13 représente un organigramme d'exécution du DES dans une variante du troisième mode d'application;

- la figure 14 représente un schéma-bloc simplifié d'une carte à puce comportant un composant électronique dans lequel le procédé de contre-mesure selon l'invention est mis en oeuvre.

L'algorithme cryptographique à clé secrète DES (dans la suite on parlera plus simplement du DES ou de l'algorithme DES) comporte 16 tours de calcul, notés T1 à T16, comme représenté sur les figures 3 et 4.

Le DES débute par une permutation initiale IP sur le message d'entrée M (figure 3). Le message d'entrée M est un mot f de 64 bits. Après permutation, on obtient un mot e de 64 bits, que l'on coupe en deux pour former

les paramètres d'entrée L0 et R0 du premier tour (T1). L0 est un mot d de 32 bits contenant les 32 bits de poids forts du mot e. R0 est un mot h de 32 bits contenant les 32 bits de poids faibles du mot e.

La clé secrète K, qui est un mot q de 64 bits subit elle-même une permutation et une compression pour fournir un mot r de 56 bits.

Le premier tour comprend une opération EXP PERM sur le paramètre R0, consistant en une expansion et une permutation, pour fournir en sortie un mot l de 48 bits.

Ce mot l est combiné à un paramètre K1, dans une opération de type OU EXCLUSIF notée XOR, pour fournir un mot b de 48 bits. Le paramètre K1 qui est un mot m de 48 bits est obtenu du mot r par un décalage d'une position (opération notée SHIFT sur les figures 3 et 4) suivi d'une permutation et d'une compression (opération notée COMP PERM).

Le mot b est appliqué à une opération notée SBOX, en sortie de laquelle on obtient un mot a de 32 bits. Cette opération particulière sera expliquée plus en détail en relation avec les figures 5 et 6.

Le mot a subit une permutation P PERM, donnant en sortie le mot c de 32 bits.

Ce mot c est combiné au paramètre d'entrée L0 du premier tour T1, dans une opération logique de type OU EXCLUSIF, notée XOR, qui fournit en sortie le mot g de 32 bits.

Le mot h (=R0) du premier tour fournit le paramètre d'entrée L1 du tour suivant (T2) et le mot g du premier tour fournit le paramètre d'entrée R1 du tour suivant. Le mot p du premier tour fournit l'entrée r du tour suivant.

Les autres tours T2 à T16 se déroulent de façon similaire, excepté en ce qui concerne l'opération de

décalage SHIFT qui se fait sur une ou deux positions selon les tours considérés.

Chaque tour Ti reçoit ainsi en entrée les paramètres Li-1, Ri-1 et r et fournit en sortie les paramètres Li et Ri et r pour le tour suivant Ti+1.

En fin d'algorithme DES (figure 4), le message chiffré est calculé à partir des paramètres L16 et R16 fournis par le dernier tour T16.

Ce calcul du message chiffré C comprend en pratique les opérations suivantes :

- formation d'un mot e' de 64 bits en inversant la position des mots L16 et R16, puis en les concaténant;
- application de la permutation IP^{-1} inverse de celle de début de DES, pour obtenir le mot f' de 64 bits formant le message chiffré C.

L'opération SBOX est détaillée sur les figures 5 et 6. Elle comprend une table de constantes TC_0 pour fournir une donnée de sortie a en fonction d'une donnée d'entrée b.

En pratique, cette table de constantes TC_0 se présente sous la forme de huit tables de constantes élémentaires $TC_0 1$ à $TC_0 8$, chacune recevant en entrée seulement 6 bits du mot b, pour fournir en sortie seulement 4 bits du mot a.

Ainsi, la table de constante élémentaire $TC_0 1$ représentée sur la figure 6 reçoit comme donnée d'entrée, les bits b1 à b6 du mot b et fournit comme donnée de sortie les bits a1 à a4 du mot a.

En pratique ces huit tables de constantes élémentaires $TC_0 1$ à $TC_0 8$ sont mémorisées en mémoire programme du composant électronique.

Dans l'opération SBOX du premier tour T1, un bit particulier de la donnée a de sortie de la table de constante TC_0 dépend de seulement 6 bits de la donnée b appliquée en entrée, c'est à dire de seulement 6 bits de la clé secrète K et du message d'entrée (M).

Dans l'opération SBOX du dernier tour T16, un bit particulier de la donnée a de sortie de la table de constante TC_0 peut être recalculé à partir de seulement 6 bits de la clé secrète K et du message chiffré (C).

Or si on reprend le principe de l'attaque DPA, si on choisit un ou des bits de la donnée de sortie a comme bits cibles, il suffit de faire une hypothèse sur 6 bits de la clé K, pour prédire la valeur du ou des bits cibles pour un message d'entrée (M) ou de sortie (C) donné. En d'autres termes, pour le DES, il suffit de faire une hypothèse sur une sous-clé de 6 bits.

Dans une attaque DPA sur un tel algorithme pour un ensemble de bits cibles donné issu d'une table de constantes élémentaire donnée, on a donc à discriminer une hypothèse de sous-clé juste parmi 64 possibles.

Ainsi, à partir des bits de sortie des huit tables de constantes élémentaires TC_01 à TC_08, on peut découvrir jusqu'à 8x6=48 bits de la clé secrète, en faisant des attaques DPA sur des bits cibles correspondants.

Dans le DES, on trouve donc des instructions critiques au sens des attaques DPA au début de l'algorithme et à la fin.

Au début de l'algorithme DES, les données qui peuvent être prédites à partir d'un message d'entrée M et d'une hypothèse de sous-clé, sont les données a et g calculées dans le premier tour (T1).

La donnée a du premier tour T1 (figure 3) est la donnée de sortie de l'opération SBOX du tour considéré. La donnée g est calculée à partir de la donnée a, par permutation (P PERM) et opération OU EXCLUSIF avec le paramètre d'entrée L0.

En fait, la donnée c du premier tour, est une donnée dérivée de la donnée a du premier tour. La donnée dérivée c correspond à une simple permutation de bits de la donnée a.

La donnée l du deuxième tour est une donnée dérivée de la donnée g du premier tour, car elle correspond à une permutation des bits du mot g, certains bits du mot g étant en outre dupliqués.

Connaissant a et g, on peut aussi connaître ces données dérivées.

Les instructions critiques du début de l'algorithme sont les instructions critiques qui manipulent soit la donnée que l'on peut prédire, comme la donnée a ou la donnée g du premier tour, soit une donnée dérivée.

Les instructions critiques manipulant la donnée a du premier tour T1 ou la donnée dérivée c sont ainsi les instructions de fin de l'opération SBOX, de l'opération P PERM et de début de l'opération XOR du premier tour T1.

Les instructions critiques manipulant la donnée g ou des données dérivées sont toutes les instructions de fin d'opération XOR de fin du premier tour T1 jusqu'aux instructions de début d'opération SBOX du deuxième tour T2, et les instructions de début de l'opération XOR en fin du troisième tour T3 (L2 = h(T2) = g(T1)).

En fin d'algorithme DES, les données qui peuvent être prédites à partir d'un message chiffré C et d'une hypothèse de sous-clé, sont la donnée a du seizième tour T16 et la donnée L15 égale au mot h du quatorzième tour T14.

Les instructions critiques manipulant la donnée a du seizième tour ou des données dérivées sont les instructions du seizième tour de fin d'opération SBOX, de l'opération de permutation P PERM et de début d'opération XOR.

Pour la donnée L15, les instructions critiques manipulant cette donnée ou des données dérivées sont toutes les instructions depuis les instructions de fin d'opération XOR en fin du quatorzième tour T14, jusqu'aux instructions de début d'opération SBOX du

quinzième tour T15, plus les instructions de début d'opération XOR en fin de seizième tour T16.

Le procédé de contre-mesure selon l'invention appliqué à cet algorithme DES consiste à rendre imprédictible chacune des données manipulées par les instructions critiques. Ainsi, quel que soit le ou les bits cibles utilisés, le signal DPA(t) sera toujours nul.

En ce qui concerne l'application du procédé de contre-mesure selon l'invention à l'algorithme DES, il faut donc appliquer la contre-mesure aux instructions critiques de début de DES et aux instructions critiques de fin de DES, pour être totalement protégé.

Dans le DES, toutes les données manipulées par des instructions critiques sont une donnée de sortie ou des données dérivées d'une donnée de sortie d'une opération SBOX.

En effet, en début de DES, les données qui peuvent être prédites sont les données a et g du premier tour T1. La donnée a est la donnée de sortie de l'opération SBOX du premier tour. La donnée g est calculée à partir de la donnée a, puisque g = P PERM(a) XOR L0. g est donc une donnée dérivée de la donnée de sortie a de l'opération SBOX du premier tour. Ainsi, toutes les données manipulées par les instructions critiques de début de DES découlent directement ou indirectement de la donnée de sortie a de l'opération SBOX du premier tour.

En ce qui concerne la fin de DES, les données qui peuvent être prédites sont la donnée a du seizième tour T16 et la donnée g du quatorzième tour T14, g étant égale à L15.

La donnée a est la donnée de sortie de l'opération SBOX du seizième tour T16.

Quant à la donnée L15, elle se calcule, dans l'exécution normale de l'algorithme DES, à partir de la

donnée de sortie a de l'opération SBOX du quatorzième tour T14 : L15 = P PERM(a) XOR L14.

Si on rend imprédictibles les données de sortie a de ces opérations SBOX particulières, on rend aussi imprédictibles toutes les données dérivées : on rend donc imprédictibles toutes les données manipulées par les instructions critiques de l'algorithme DES. Si on considère que ces opérations SBOX constituent des premiers moyens pour fournir une donnée de sortie S=a à partir d'une donnée d'entrée E=b, le procédé de contre-mesure appliqué à l'algorithme DES consiste à utiliser d'autres moyens pour rendre imprédictibles la donnée de sortie, en sorte que cette donnée de sortie et/ou des données dérivées manipulées par les instructions critiques soient toutes imprédictibles.

Ces autres moyens peuvent comprendre différents moyens. Ils sont calculés à partir des premiers moyens en appliquant un OU exclusif avec une valeur aléatoire ou une valeur aléatoire dérivée sur l'une et/ou sur l'autre des données d'entrée et de sortie des premiers moyens.

L'utilisation de cette valeur aléatoire est telle que le résultat en sortie, c'est à dire, le message chiffré reste juste.

La figure 7 représente un premier mode de réalisation de l'invention. Dans ce mode de réalisation, on répartit les seize tours de l'algorithme DES en quatre groupes G1 à G4 de quatre tours successifs. Le groupe G1 comprend ainsi les tours T1 à T4, le groupe G2, les tours T5 à T8, le groupe G3, les tours T9 à T12 et le groupe G4, les tours T13 à T16.

Dans une exécution classique de l'algorithme DES, on a vu que chaque tour comprend l'utilisation de premiers moyens TC_0 dans une opération SBOX.

Dans le premier mode d'application du procédé de contre-mesure, on calcule d'autres moyens en faisant un OU EXCLUSIF avec une valeur aléatoire u et/ou avec une valeur dérivée e(p(u)) sur l'une et/ou l'autre des données d'entrée et de sortie des premiers moyens TC_0. Puis on applique une séquence SEQA d'exécution identique sur chaque groupe, qui consiste à utiliser ces autres moyens calculés.

Selon l'invention, on utilise une valeur aléatoire u qui est une donnée de 32 bits. On peut par exemple tirer une valeur aléatoire de 32 bits, ou bien tirer une valeur aléatoire de 4 bits et les recopier 8 fois pour obtenir la valeur aléatoire u sur 32 bits.

On calcule alors la variable dérivée égale à e(p(u)), où p(u) correspond au résultat de l'opération P PERM appliquée sur la valeur u et où e(p(u)) est le résultat de l'opération EXP PERM appliquée à la valeur p(u).

On peut alors calculer les autres moyens utilisés dans l'invention.

Dans l'exemple représenté en référence à la figure 7, ces autres moyens comprennent des deuxièmes moyens TC_2 et des troisièmes moyens TC_1.

Les deuxièmes moyens TC_2 sont utilisés dans le deuxième tour et l'avant-dernier tour de chaque groupe : c'est à dire, dans T2, T3 de G1, T6, T7 de G2, T10, T11 de G3 et T14 et T15 de G4.

Les deuxièmes moyens TC_2 sont utilisés dans le deuxième tour et l'avant-dernier tour de chaque groupe : c'est à dire, dans T2, T3 de G1, T6, T7 de G2, T10, T11 de G3 et T14 et T15 de G4.

Les deuxièmes moyens TC_2 sont calculés en appliquant un OU EXCLUSIF avec la variable aléatoire dérivée e(p(u)) sur la donnée d'entrée E et en appliquant un OU EXCLUSIF avec la valeur aléatoire u

sur la donnée de sortie S des premiers moyens TC_0, ce qui peut s'écrire : $TC_2=(E\oplus e(p(u)), S\oplus u)$.

Les troisièmes moyens TC_1 sont utilisés dans le premier tour et le dernier tour de chaque groupe. C'est à dire dans T1, T4 de G1, T5, T8 de G2, T9, T12 de G3 et T13, T16 de G4.

Les troisièmes moyens TC_1 sont calculés en appliquant un OU EXCLUSIF avec la variable aléatoire u sur la donnée de sortie S des premiers moyens TC_0, ce qui peut s'écrire : $TC_1=(E, S\oplus u)$.

Le programme de calcul consiste alors au début de l'exécution de l'algorithme, à tirer une valeur aléatoire u, dans l'exemple sur 4 bits, à calculer la variable aléatoire dérivée e(p(u)), puis à calculer les différents moyens utilisés dans la séquence d'exécution SEQA. Dans l'exemple, il faut calculer les deuxièmes et troisièmes moyens TC_2 et TC_1.

On obtient, à la sortie de chaque groupe, le résultat juste pour les paramètres de sortie. Ainsi, les paramètres de sortie L4 et R4 du premier groupe G1, L8 et R8 du deuxième groupe G2, L12 et R12 du troisième groupe G3, L16 et R16 du quatrième groupe G4 sont justes quelle que soit la variable aléatoire tirée.

Quand on a effectué tous les tours, on obtient les paramètres justes L16 et R16 qui vont permettre de calculer le message chiffré C juste.

Par contre, à l'intérieur des groupes, certains résultats intermédiaires n'ont pas les mêmes valeurs selon la séquence utilisée, mais des valeurs correspondant à l'opération OU EXCLUSIF avec la valeur aléatoire u ou avec la valeur aléatoire dérivée e(p(u)), comme on va le montrer par référence aux figures 3 et 8.

La figure 8 montre l'organigramme détaillé des quatre tours T1, T2, T3 et T4 du premier groupe G1, dans la séquence SEQA d'exécution selon l'invention.

Dans cette séquence, le tour T1 utilise les troisièmes moyens TC_1. En sortie de l'opération SBOX, on obtient donc la donnée modifiée aléatoirement $a \oplus u$ (Figure 8), au lieu de la donnée a selon la séquence normale du DES, c'est à dire sans contre-mesure (Figure 3).

Avec la séquence SEQA d'exécution selon l'invention, l'opération P PERM du premier tour T1 qui est une simple permutation va donc également fournir en sortie une donnée modifiée aléatoirement égale à $c \oplus p(u)$.

La donnée qui est obtenue par l'opération XOR entre une donnée $c \oplus p(u)$ et la donnée L0, va aussi fournir en sortie une donnée modifiée aléatoirement $g \oplus p(u)$. Cette donnée appliquée à l'opération EXP PERM va fournir en sortie la donnée modifiée aléatoirement notée $l \oplus e(p(u))$.

Ainsi, avec les troisièmes moyens TC_1 du tour T1 on obtient toutes les données modifiées aléatoirement suivantes :

- dans le tour T1 : $a \oplus u$, $c \oplus p(u)$, $g \oplus p(u)$;
- dans le tour T2 : $R1 \oplus p(u)$, $h \oplus p(u)$, $l \oplus e(p(u)$, $b \oplus e(p(u)$;
- dans le tour T3 : $L2 \oplus p(u)$.

On arrive alors aux deuxièmes moyens TC_2 utilisés dans le tour T2. D'après leur définition : $E \oplus e(p(u)$, $S \oplus u$, en appliquant en entrée la donnée modifiée aléatoirement $b \oplus e(p(u)$, on obtient en sortie la donnée modifiée aléatoirement $a \oplus u$. En conduisant ce raisonnement jusqu'à la fin du tour T4, et en remarquant que $p(u) \oplus p(u)=0$, on obtient en sortie du tour T4, les données L4, R4 non modifiées.

En outre, on constate que pour toutes les instructions critiques de début de DES, les instructions critiques vont manipuler, des données modifiées de manière aléatoire.

Avec un tel procédé de contre-mesure, on doit prévoir en début de DES le tirage de la valeur aléatoire u et le calcul des moyens utilisés dans la séquence d'exécution SEQA. Ces moyens calculés à chaque exécution du DES, sont mémorisés, le temps de l'exécution, en mémoire de travail, les premiers moyens TC_0 qui servent au calcul étant eux mémorisés en mémoire programme.

En revenant à la figure 7, on pourra noter, que l'on n'a pas besoin de contre-mesure dans les groupes du milieu G2 et G3, puisqu'ils ne contiennent pas d'instructions critiques au sens attaque DPA. On pourrait donc n'appliquer la séquence d'exécution SEQA du procédé de contre-mesure qu'au premier et au dernier groupe G1 et G4. Il suffirait d'utiliser ensuite les premiers moyens (TC_0) dans les groupes G2 et G3.

Mais le fait de d'appliquer le procédé de contre-mesure à tous les groupes donne une cohérence à l'ensemble.

Ainsi, on applique la séquence SEQA à chacun des groupes G1 à G4.

Un deuxième mode de réalisation du procédé de contre-mesure est représentée sur la figure 9. Ce deuxième mode de réalisation est en fait une variante du premier.

L'intérêt de cette variante est de n'utiliser comme autres moyens dans la séquence SEQA, que les deuxièmes moyens TC_2. En effet, on a vu que les différents moyens TC_0, TC_1, TC_2 correspondent en pratique à des tables de constantes comprenant chacune huit tables de constantes élémentaires, qu'il faut recalculer en ce qui concerne les moyens TC_1 et TC_2 à chaque nouvelle exécution du DES, et garder en mémoire de travail.

Cette variante consiste donc à utiliser uniquement les deuxièmes moyens TC_2 dans la séquence SEQA. Pour cela, on prévoit dans le programme de calcul des

premiers et derniers tours de chaque groupe, une opération OU EXCLUSIF supplémentaire CP avec la variable aléatoire dérivée e(p(u)), pour obtenir en entrée des deuxièmes moyens la donnée b⊕e(p(u)). On note cette opération CP(e(p(u))) sur les figures. Si on se reporte à la figure 10 représentant l'organigramme détaillé de la séquence SEQA d'exécution des quatre tours T1 à T4 du premier groupe G1, il s'agit donc d'appliquer en entrée de l'opération SBOX des tours T1 et T4 une variable b⊕e(p(u)). L'opération supplémentaire CP plus les deuxièmes moyens TC_2 équivalent aux troisièmes moyens TC_1 utilisés dans le premier mode de réalisation de l'invention.

On y gagne en temps de calcul, car l'opération CP n'est exécutée que deux fois dans un groupe, soit 8 fois pour une séquence SEQA complète sur les quatre groupes, alors que le calcul d'une table nécessite de faire cette opération b⊕e(p(u)) pour toutes les données d'entrée de cette table.

On notera que l'opération OU EXCLUSIF supplémentaire CP avec la variable e(p(u)) peut être placée en divers endroits des premiers et derniers tours, soit entre l'opération EXP PERM et l'opération XOR ou entre l'opération XOR et l'opération SBOX.

On peut aussi remarquer que l'on peut utiliser une opération OU EXCLUSIF supplémentaire CP avec la variable aléatoire dérivée p(u), en plaçant cette opération supplémentaire CP(p(u) avant l'opération EXP PERM. On obtient en sortie l⊕e(p(u)), et donc on aura ensuite b⊕e(p(u)).

Dans tous ces cas de figures, on obtient la donnée b⊕e(p(u)) en entrée de l'opération SBOX.

La figure 11 représente un troisième exemple de réalisation d'un procédé de contre-mesure selon l'invention.

Dans ce mode de réalisation, on forme un premier groupe G1 avec les trois premiers tours T1, T2, T3 et un autre groupe G4 avec les trois derniers tours T14, T15, T16. On applique sur chaque groupe la séquence d'exécution SEQA avec les autres moyens pour certains tours au moins.

Pour les autres tours non compris dans les groupes, c'est à dire pour les tours T4 à T13, on applique les premiers moyens TC_0.

En sortie de chaque groupe G1, G4, on obtient le bon résultat en sortie L3, R3 et L16, R16, quelle que soit la variable aléatoire u tirée.

Les autres moyens sont dans l'exemple les troisièmes moyens TC_1 déjà vus en relation avec le premier mode de réalisation et des quatrièmes moyens TC_3.

Ces quatrièmes moyens sont calculés par rapport aux premiers moyens TC_0 en appliquant un OU EXCLUSIF sur la donnée E d'entrée, avec la variable aléatoire dérivée e(p(u)).

Ainsi, après avoir tiré la valeur aléatoire u, et calculé la variable aléatoire dérivée, on calcule les différents moyens utilisés dans la séquence d'exécution SEQA. Puis on applique cette séquence SEQA sur le premier groupe. On obtient en sortie les paramètres L3, R3. On exécute les tours suivants T4 à T13 avec les premiers moyens TC_0. En fin de tour T13, on applique la séquence SEQA sur le groupe G4. On obtient les paramètres L16, R16 qui vont servir à calculer le message chiffré C.

La figure 12 est un organigramme détaillé correspondant.

Il apparaît clairement sur cet organigramme que l'on obtient des données modifiées aléatoirement pour toutes les instructions critiques de ces tours. Les données L3 et R3 en sortie du troisième tour ne sont

pas modifiées, ce qui permet de continuer l'exécution de l'algorithme, en passant au tour T4 auquel on applique les premiers moyens TC_0 selon l'exécution normale de l'algorithme.

Sur cette figure, on peut remarquer que dans l'opération SBOX du troisième tour T3, on pourrait utiliser les premiers moyens TC_0 à la place des troisièmes moyens calculés TC_1, en prévoyant une opération OU EXCLUSIF supplémentaire CP en sortie de l'opération SBOX, pour faire un OU exclusif de la sortie avec la variable aléatoire u, pour obtenir la donnée $a \oplus u$ en entrée de l'opération XOR. C'est une solution équivalente.

La figure 13 représente un organigramme d'exécution utilisant cette variante. Pour le troisième tour des deux groupes G1 et G4, on utilise dans la séquence d'exécution SEQA, les premiers moyens TC_0 suivis en sortie de l'instruction OU EXCLUSIF supplémentaire avec la variable u, ce qui est noté $T3(TC_0, CP(u))$.

De manière générale, dans le procédé de contre-mesure selon l'invention, on peut donc prévoir dans la séquence d'exécution SEQA et pour un ou plusieurs tours, une instruction OU EXCLUSIF supplémentaire CP en entrée ou en sortie des moyens utilisés avec la variable u ou une variable aléatoire dérivée p(u) ou e(p(u)) selon les cas.

La présente invention s'applique à l'algorithme de cryptographie à clé secrète DES, pour lequel plusieurs exemples d'application non limitatifs ont été décrits. Il s'applique plus généralement à un algorithme de cryptographie à clé secrète à seize tours de calculs, dont les instructions critiques se situent parmi les instructions des trois premiers ou trois derniers tours.

Un composant électronique 1 mettant en oeuvre un procédé de contre-mesure selon l'invention dans un

algorithme de cryptographie à clé secrète DES, comprend typiquement, comme représenté sur la figure 10, un microprocesseur µP, une mémoire programme 2 et une mémoire de travail 3. Pour pouvoir gérer l'utilisation des différents moyens TC_0, TC_1, TC_2 selon l'invention, qui sont, en pratique, des tables de constantes mémorisées en mémoire programme, des moyens 4 de génération d'une valeur aléatoire entre 0 et 1, sont prévus qui, si on se reporte aux organigrammes des figures 7 et 11, fourniront la valeur aléatoire u à chaque exécution du DES. Un tel composant peut tout particulièrement être utilisé dans une carte à puce 5, pour améliorer son inviolabilité.

REVENDICATIONS

1. Procédé de contre-mesure dans un composant électronique mettant en oeuvre un algorithme cryptographique à clé secrète (K), la mise en oeuvre de l'algorithme comprenant l'utilisation de premiers moyens (TC_0) pour fournir une donnée de sortie (S) à partir d'une donnée d'entrée (E), la donnée de sortie et/ou des données dérivées étant manipulées par des instructions critiques, caractérisé en ce que le procédé de contre-mesure prévoit l'utilisation d'autres moyens (TC_1), en sorte que la donnée de sortie et les données dérivées soient imprédictibles, ces autres moyens étant obtenus des dits premier moyens par une opération de OU EXCLUSIF avec une valeur aléatoire (u) ou une valeur aléatoire dérivée (e(p(u))) sur l'une et/ou sur l'autre des données d'entrée et de sortie des dits premiers moyens.

2. Procédé de contre-mesure selon la revendication 1, la mise en oeuvre de l'algorithme comprenant seize tours de calcul (T1, ...,T16), chaque tour utilisant des premiers moyens (TC_0) pour fournir une donnée de sortie à partir d'une donnée d'entrée, la donnée de sortie et/ou des données dérivées étant manipulées par des instructions critiques dans les trois premiers (T1, T2, T3) et les trois derniers tours (T14, T15, T16), caractérisé en ce que l'on forme un groupe (G1) comprenant les trois premiers tours au moins et un autre groupe (G4) comprenant les trois derniers tours au moins, et en ce que l'on associe au premier groupe (G1) et au dernier groupe (G4) une séquence d'exécution (SEQA) utilisant les autres moyens (TC_1, TC_2) dans certains tours au moins.

3. Procédé de contre-mesure selon la revendication 2, caractérisé en ce que l'on forme quatre groupes (G1,...G4) de quatre tours successifs chacun (T1,...T4) et , en ce que l'on applique au moins au premier groupe (G1) et au dernier groupe (G4) la dite séquence d'exécution (SEQA).

4. Procédé de contre-mesure selon la revendication 3, caractérisé en ce que la dite séquence (SEQA) est exécutée dans chacun des groupes (G1,...G4).

5. Procédé de contre-mesure selon la revendication 2, caractérisé en ce que la dite séquence d'exécution (SEQA) est appliquée à un premier groupe (G1) formé des trois premiers tours (T1, T2, T3) et à un dernier groupe formé des trois derniers tours (T14, T15, T16).

6. Procédé de contre-mesure selon l'une quelconque des revendications précédentes, caractérisé en ce que chaque exécution de l'algorithme comprend le tirage d'une valeur aléatoire (u), et le calcul des autres moyens.

7. Procédé de contre-mesure selon l'une quelconque des revendications précédentes, caractérisé en ce que les différents moyens sont des tables de constantes.

8. Procédé de contre-mesure selon l'une quelconque des revendications précédentes, caractérisé en ce que les différents moyens sont utilisés en combinaison avec une opération OU exclusif supplémentaire (CP) avec la valeur aléatoire ou une valeur dérivée (p(u), e(p(u))).

9. Composant électronique de sécurité mettant en oeuvre le procédé de contre-mesure selon l'une quelconque des revendications précédentes, caractérisé en ce que les premiers moyens (TC_0), pour fournir une donnée de sortie à partir d'une donnée d'entrée sont fixés en mémoire programme (1) du dit composant, les autres moyens (TC_1, TC_2) étant calculés à chaque nouvelle exécution de l'algorithme et mémorisés en mémoire de travail (3) et en ce qu'il comprend des moyens (4) de génération d'une valeur aléatoire (u) pour calculer les dits autres moyens.

10. Carte à puce comprenant un composant électronique de sécurité selon la revendication 9.

FIG.1

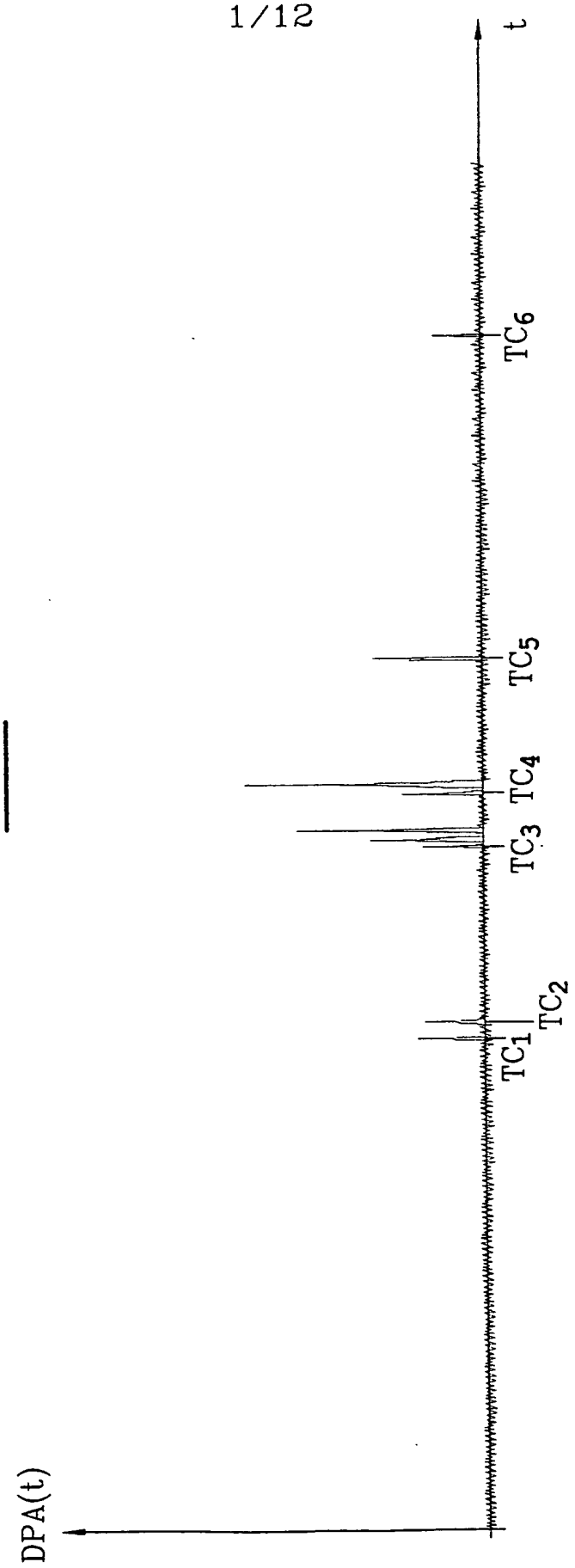
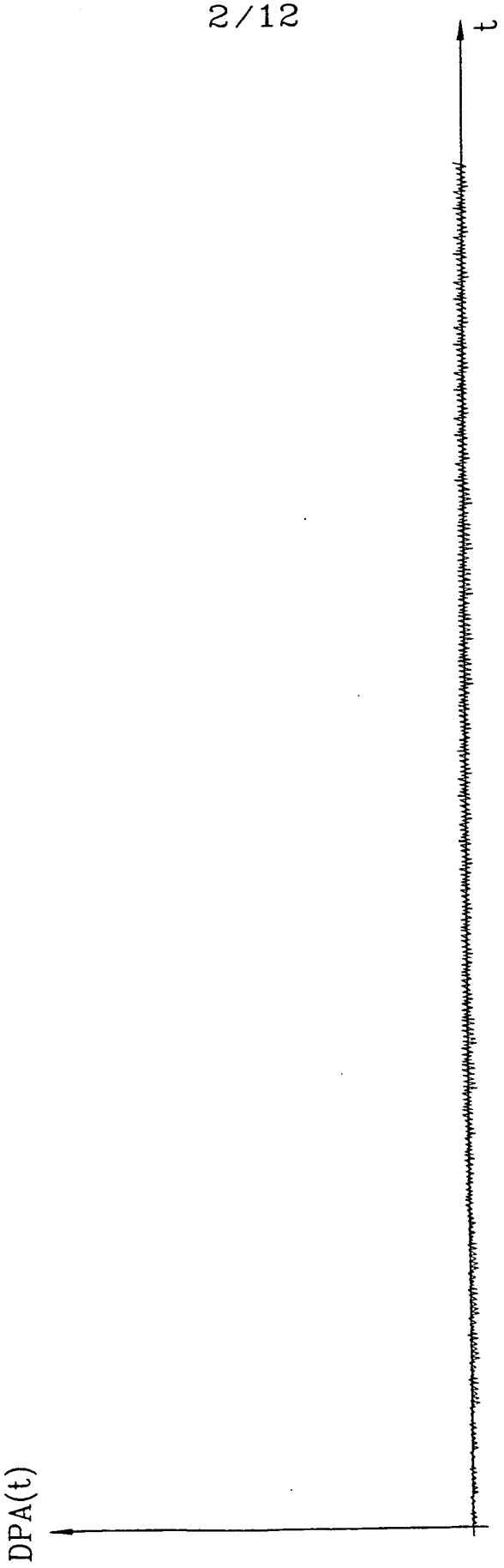


FIG.2



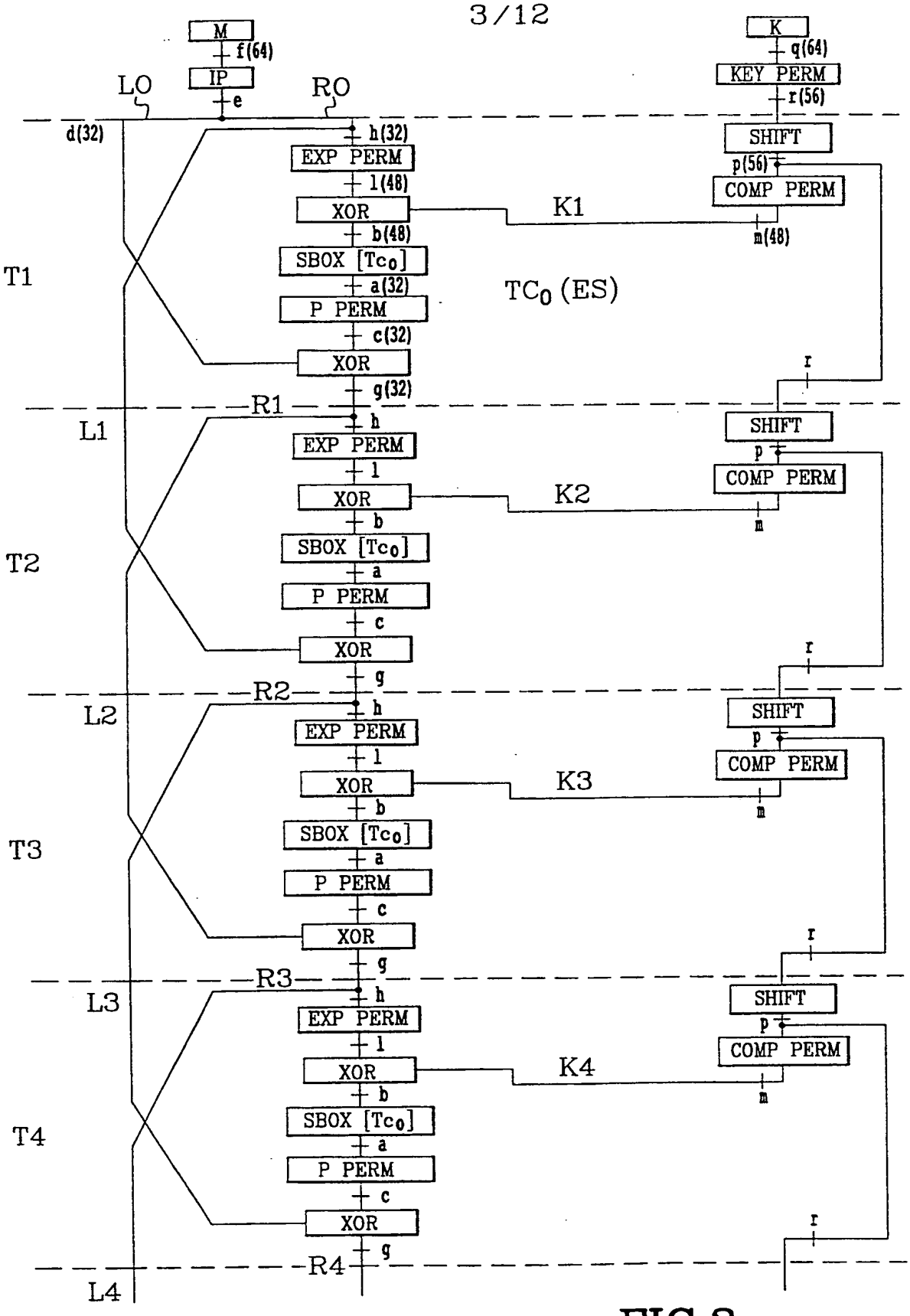

M
f(64)
IP
e
LO
RO
d(32)
h(32)
EXP PERM
l(48)
XOR
b(48)
SBOX [Tc0]
a(32)
P PERM
c(32)
XOR
g(32)
K
q(64)
KEY PERM
r(56)
SHIFT
p(56)
COMP PERM
m(48)
K1
TC0 (ES)
T1
R1
L1
T2
R2
L2
T3
R3
L3
T4
R4
L4
T5
K2
K3
K4

FIG.3

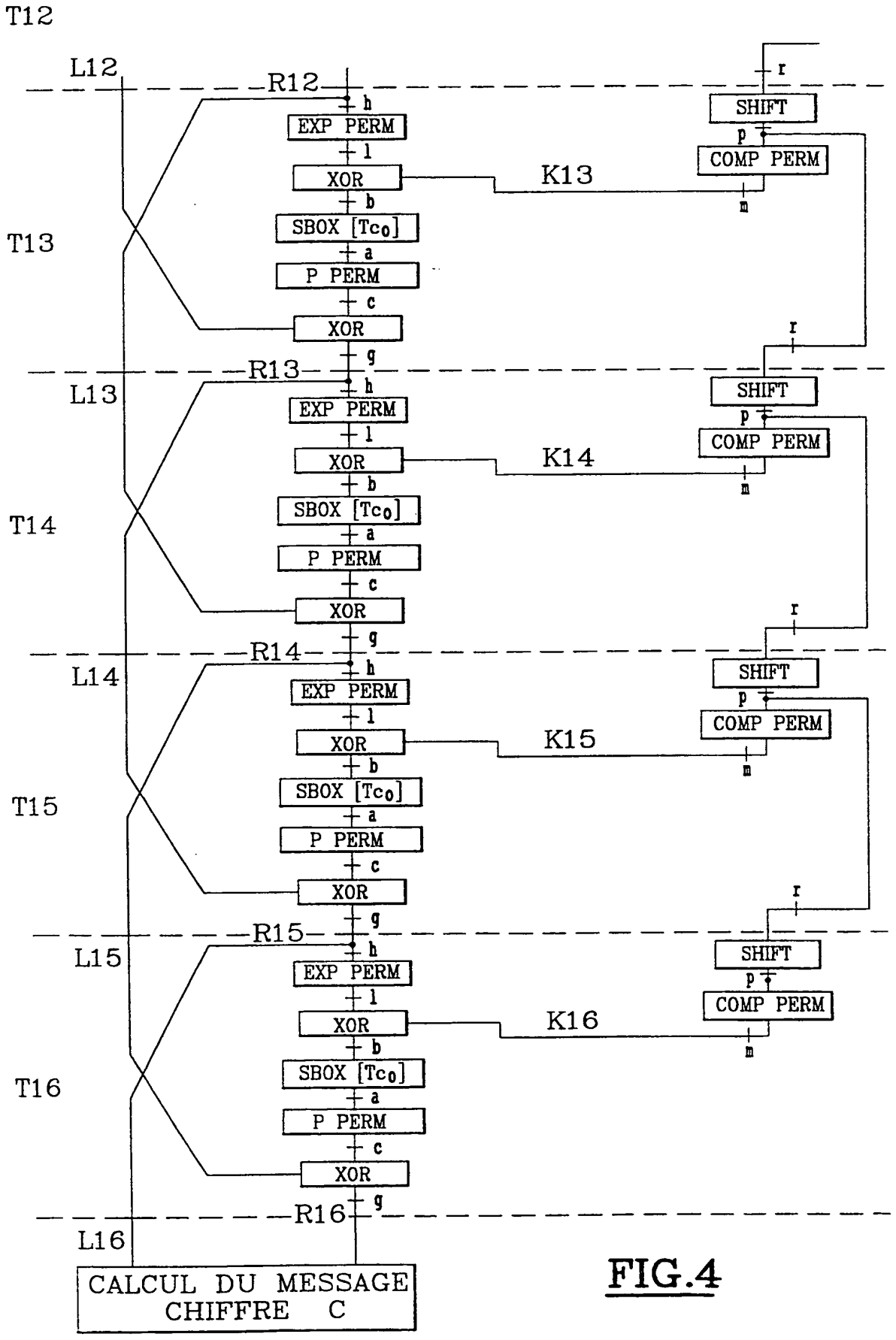

T12
L12
R12
h
EXP PERM
l
XOR
K13
b
SBOX [Tc0]
T13
a
P PERM
c
XOR
g
r
SHIFT
p
COMP PERM
m
R13
L13
K14
T14
R14
L14
K15
T15
R15
L15
K16
T16
R16
L16
CALCUL DU MESSAGE CHIFFRE C

FIG.4

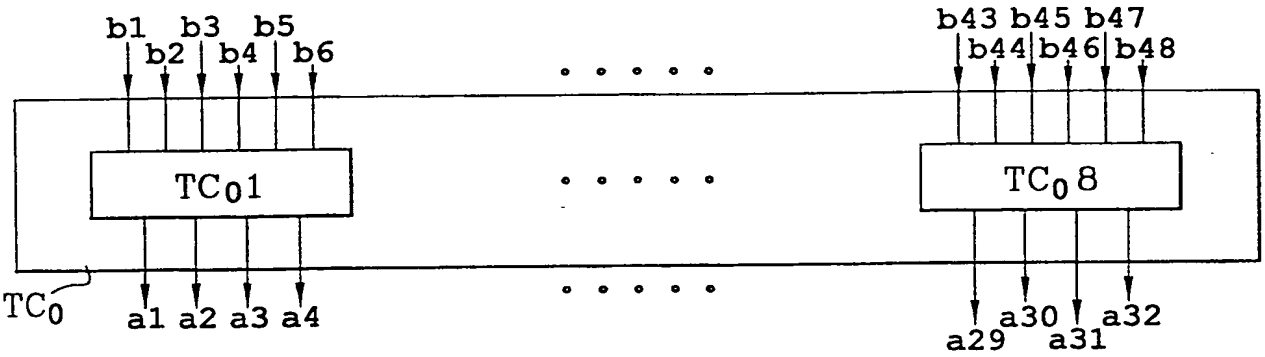


FIG.5

FIG.6

TC_0 1

E=b1b2b3b4b5b6	S=a1a2a3a4
000000	1101
000001	0101
⋮	⋮
111111	1010

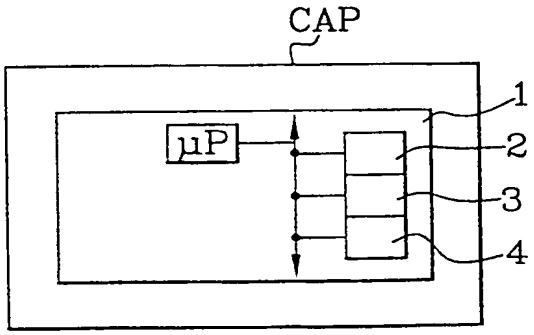


FIG.14

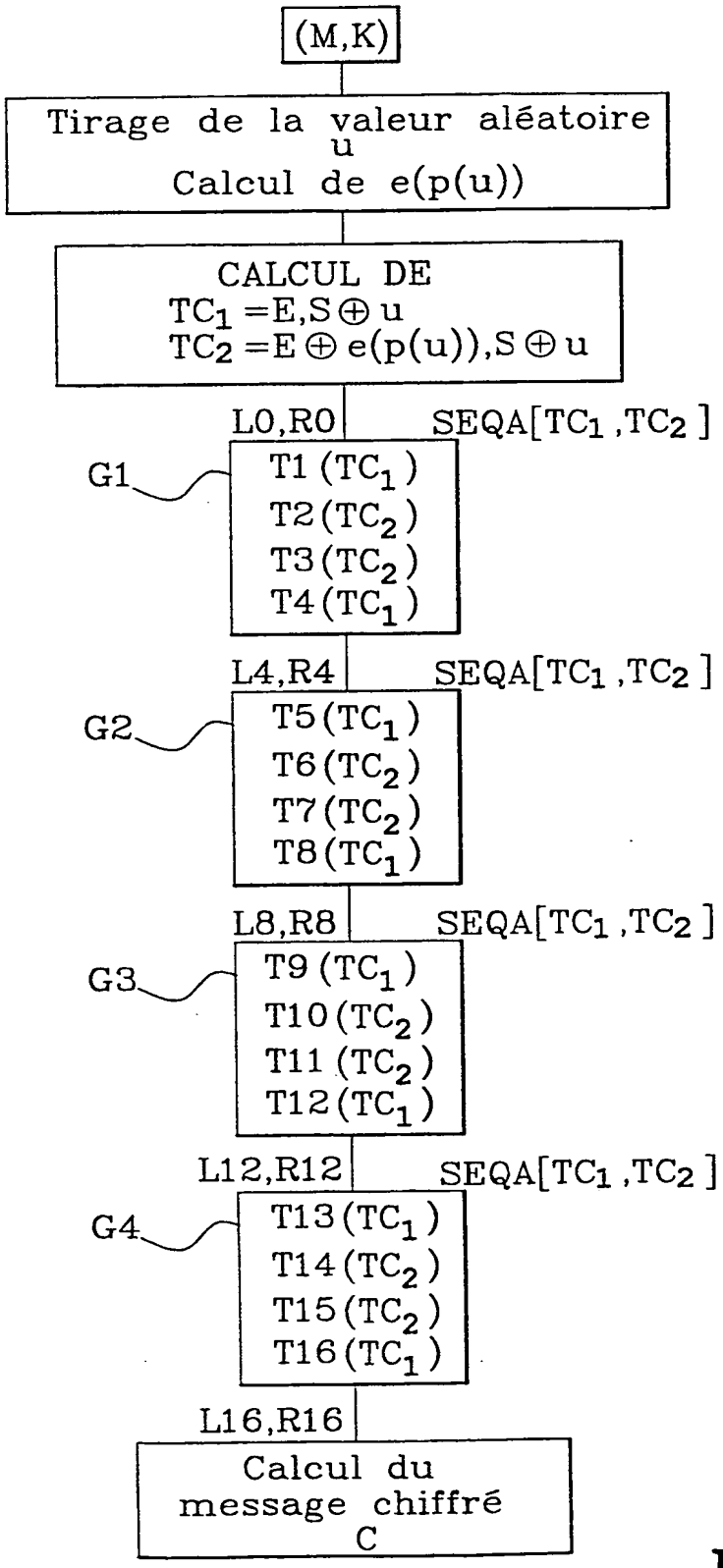

(M,K)
Tirage de la valeur aléatoire u
Calcul de e(p(u))
CALCUL DE
TC1 =E,S⊕u
TC2 =E⊕e(p(u)),S⊕u
L0,R0
SEQA[TC1 ,TC2]
G1
T1 (TC1)
T2(TC2)
T3(TC2)
T4(TC1)
L4,R4
SEQA[TC1 ,TC2]
G2
T5(TC1)
T6(TC2)
T7(TC2)
T8(TC1)
L8,R8
SEQA[TC1 ,TC2]
G3
T9 (TC1)
T10(TC2)
T11 (TC2)
T12(TC1)
L12,R12
SEQA[TC1 ,TC2]
G4
T13(TC1)
T14(TC2)
T15(TC2)
T16(TC1)
L16,R16
Calcul du message chiffré C

FIG.7

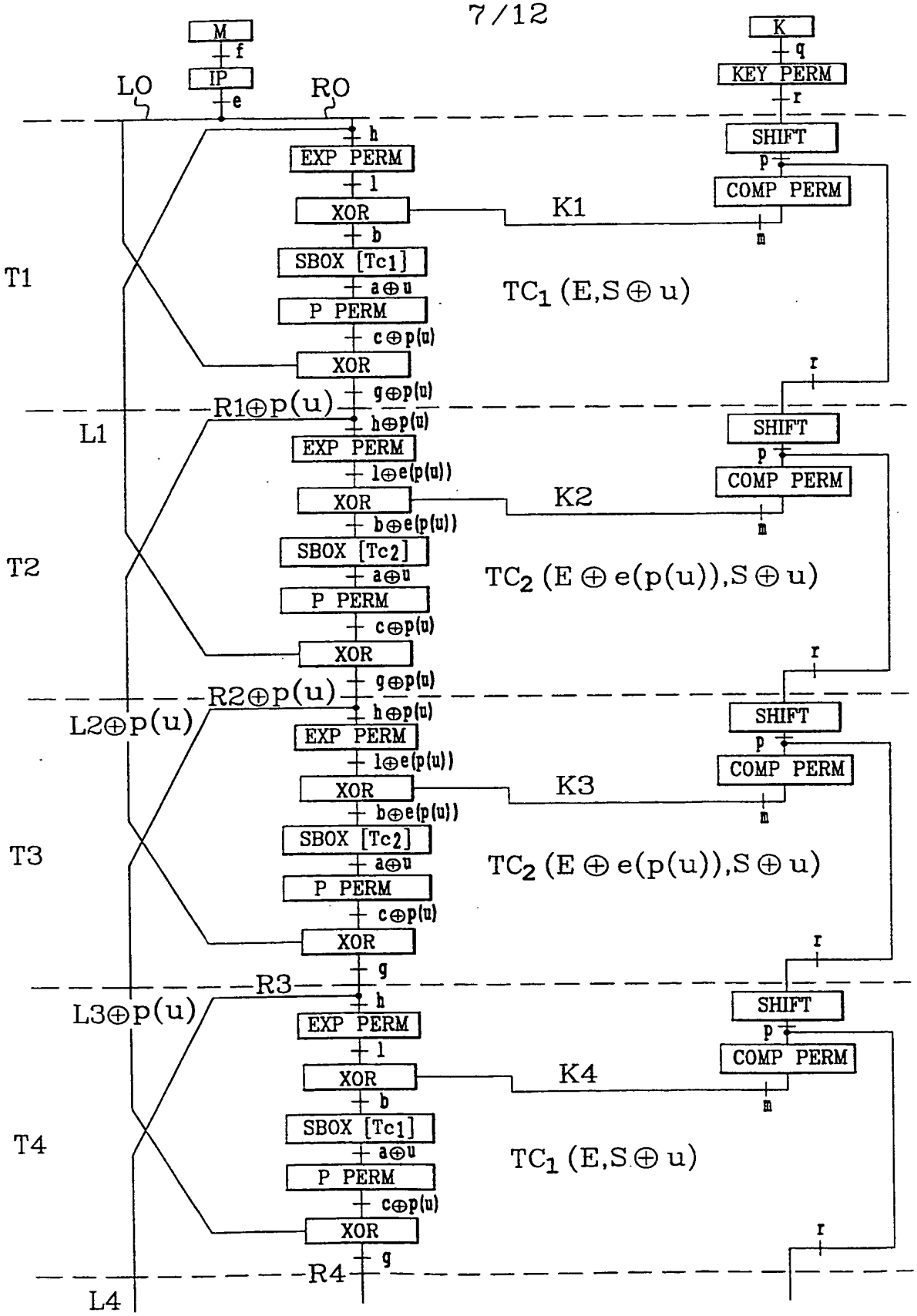

M
f
IP
e
L0
R0
K
q
KEY PERM
r
SHIFT
p
COMP PERM
m
K1
h
EXP PERM
l
XOR
b
SBOX [Tc1]
a⊕u
P PERM
c⊕p(u)
XOR
g⊕p(u)
T1
TC1 (E,S⊕u)
R1⊕p(u)
L1
h⊕p(u)
EXP PERM
l⊕e(p(u))
XOR
K2
b⊕e(p(u))
SBOX [Tc2]
a⊕u
P PERM
c⊕p(u)
XOR
g⊕p(u)
T2
TC2 (E⊕e(p(u)),S⊕u)
R2⊕p(u)
L2⊕p(u)
h⊕p(u)
EXP PERM
l⊕e(p(u))
XOR
K3
b⊕e(p(u))
SBOX [Tc2]
a⊕u
P PERM
c⊕p(u)
XOR
g
T3
TC2 (E⊕e(p(u)),S⊕u)
R3
L3⊕p(u)
h
EXP PERM
l
XOR
K4
b
SBOX [Tc1]
a⊕u
P PERM
c⊕p(u)
XOR
g
T4
TC1 (E,S⊕u)
R4
L4

FIG.8

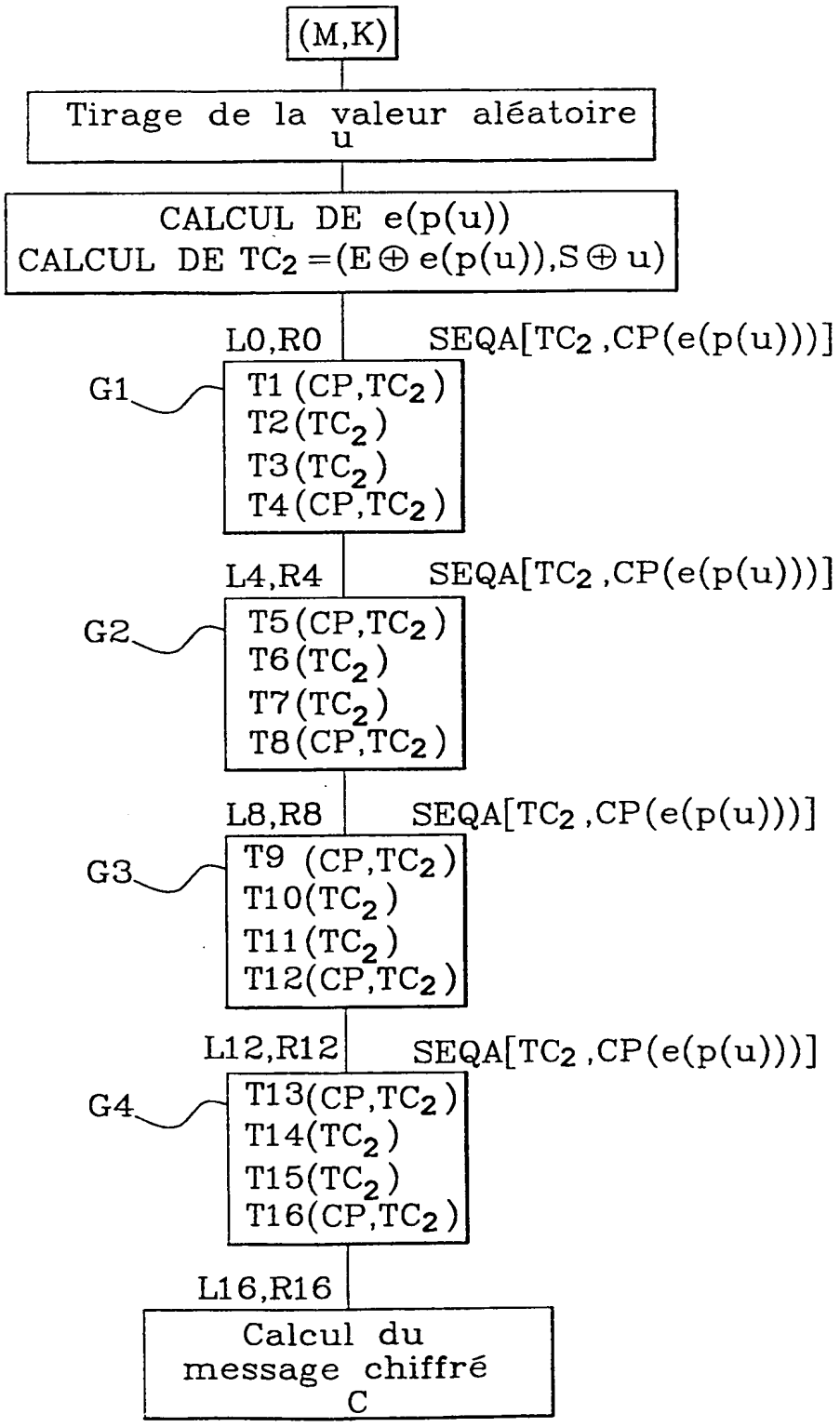
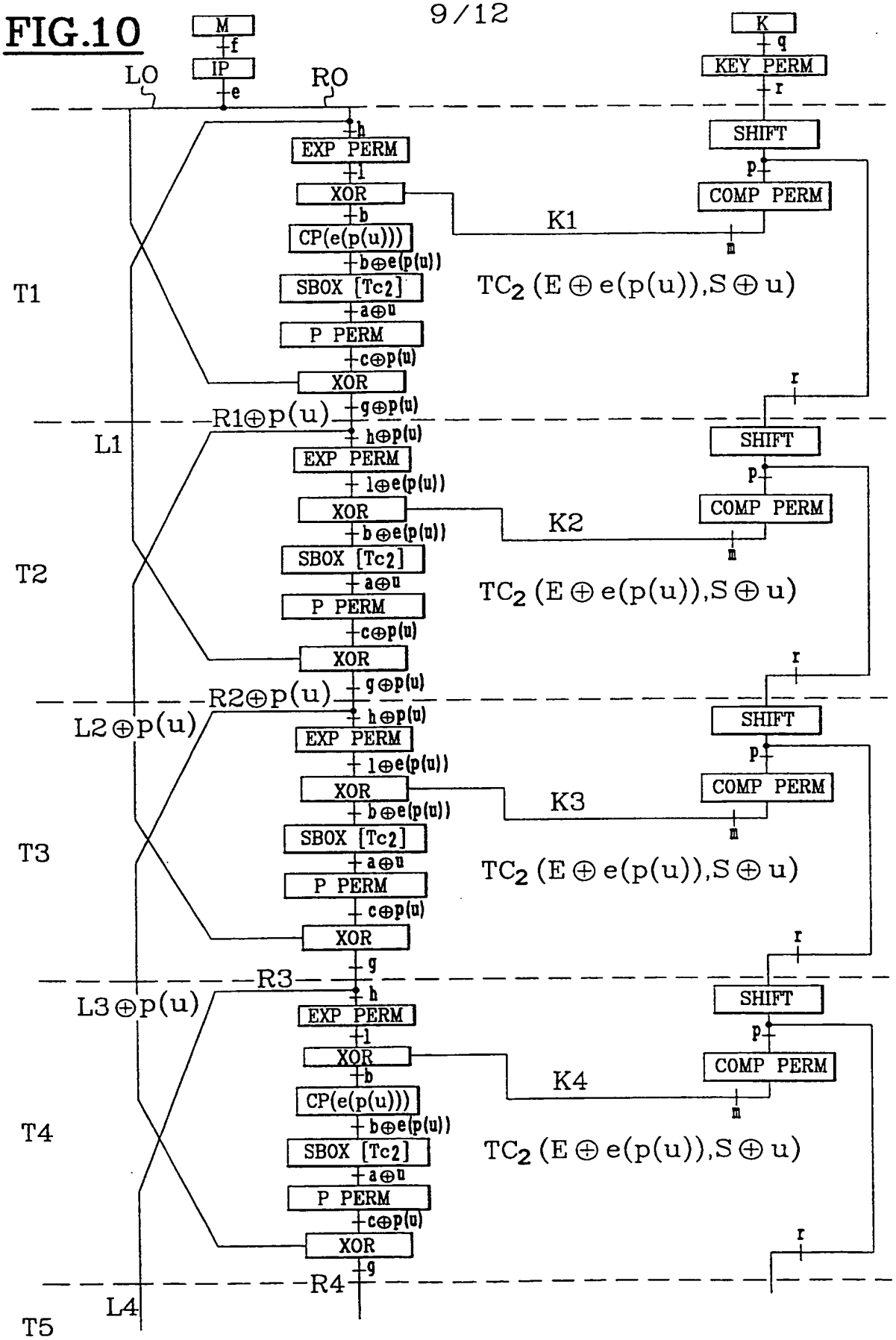

(M,K)
Tirage de la valeur aléatoire u
CALCUL DE e(p(u))
CALCUL DE TC2 =(E⊕e(p(u)),S⊕u)
L0,R0
SEQA[TC2 ,CP(e(p(u)))]
G1
T1 (CP,TC2)
T2(TC2)
T3(TC2)
T4(CP,TC2)
L4,R4
SEQA[TC2 ,CP(e(p(u)))]
G2
T5(CP,TC2)
T6(TC2)
T7(TC2)
T8(CP,TC2)
L8,R8
SEQA[TC2 ,CP(e(p(u)))]
G3
T9 (CP,TC2)
T10(TC2)
T11(TC2)
T12(CP,TC2)
L12,R12
SEQA[TC2 ,CP(e(p(u)))]
G4
T13(CP,TC2)
T14(TC2)
T15(TC2)
T16(CP,TC2)
L16,R16
Calcul du message chiffré C

FIG.9

FIG.10


M
f
IP
e
LO
RO
K
q
KEY PERM
r
SHIFT
p
COMP PERM
m
K1
h
EXP PERM
l
XOR
b
CP(e(p(u)))
b⊕e(p(u))
SBOX [Tc2]
a⊕u
P PERM
c⊕p(u)
XOR
g⊕p(u)
TC2 (E⊕e(p(u)),S⊕u)
T1
R1⊕p(u)
L1
h⊕p(u)
l⊕e(p(u))
K2
T2
R2⊕p(u)
L2⊕p(u)
K3
T3
R3
L3⊕p(u)
g
K4
T4
R4
L4
T5

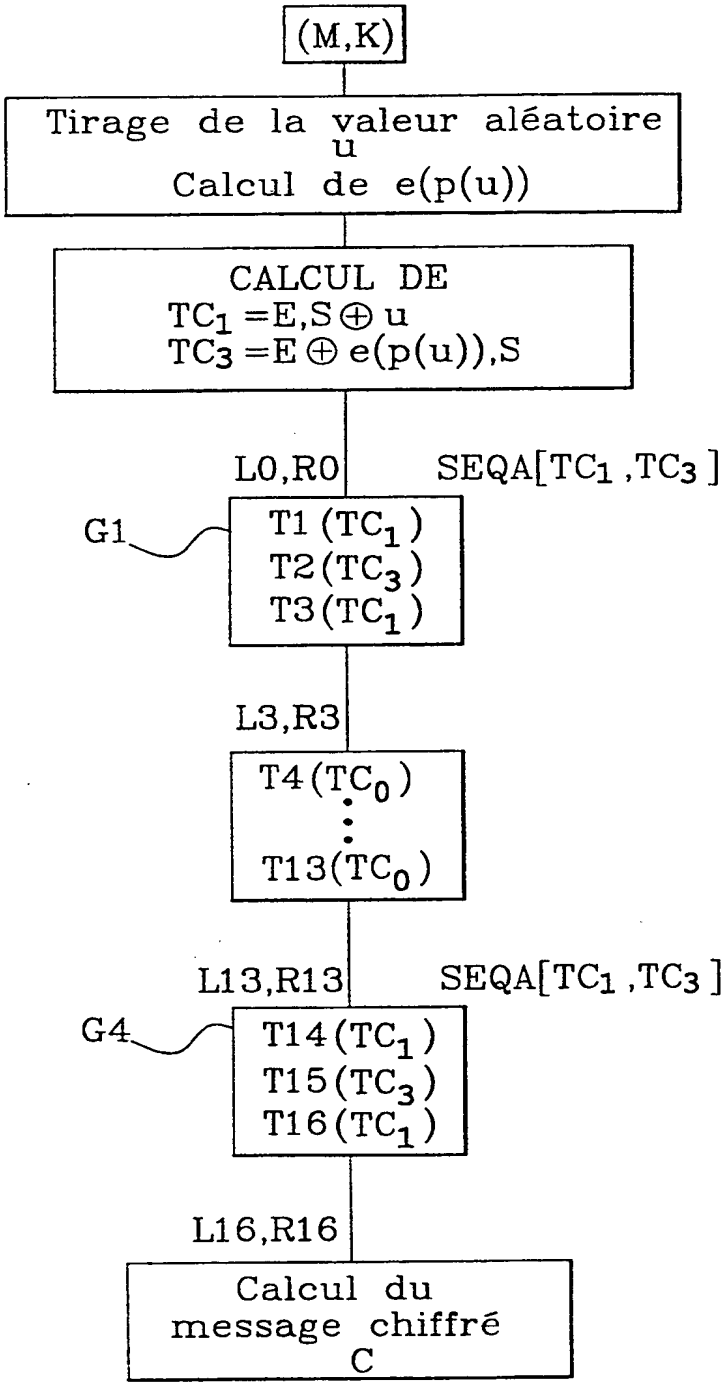

(M,K)
Tirage de la valeur aléatoire u
Calcul de e(p(u))
CALCUL DE
TC1 =E,S⊕u
TC3 =E⊕e(p(u)),S
L0,R0
SEQA[TC1 ,TC3]
G1
T1(TC1)
T2(TC3)
T3(TC1)
L3,R3
T4(TC0)
T13(TC0)
L13,R13
SEQA[TC1 ,TC3]
G4
T14(TC1)
T15(TC3)
T16(TC1)
L16,R16
Calcul du message chiffré C

FIG.11

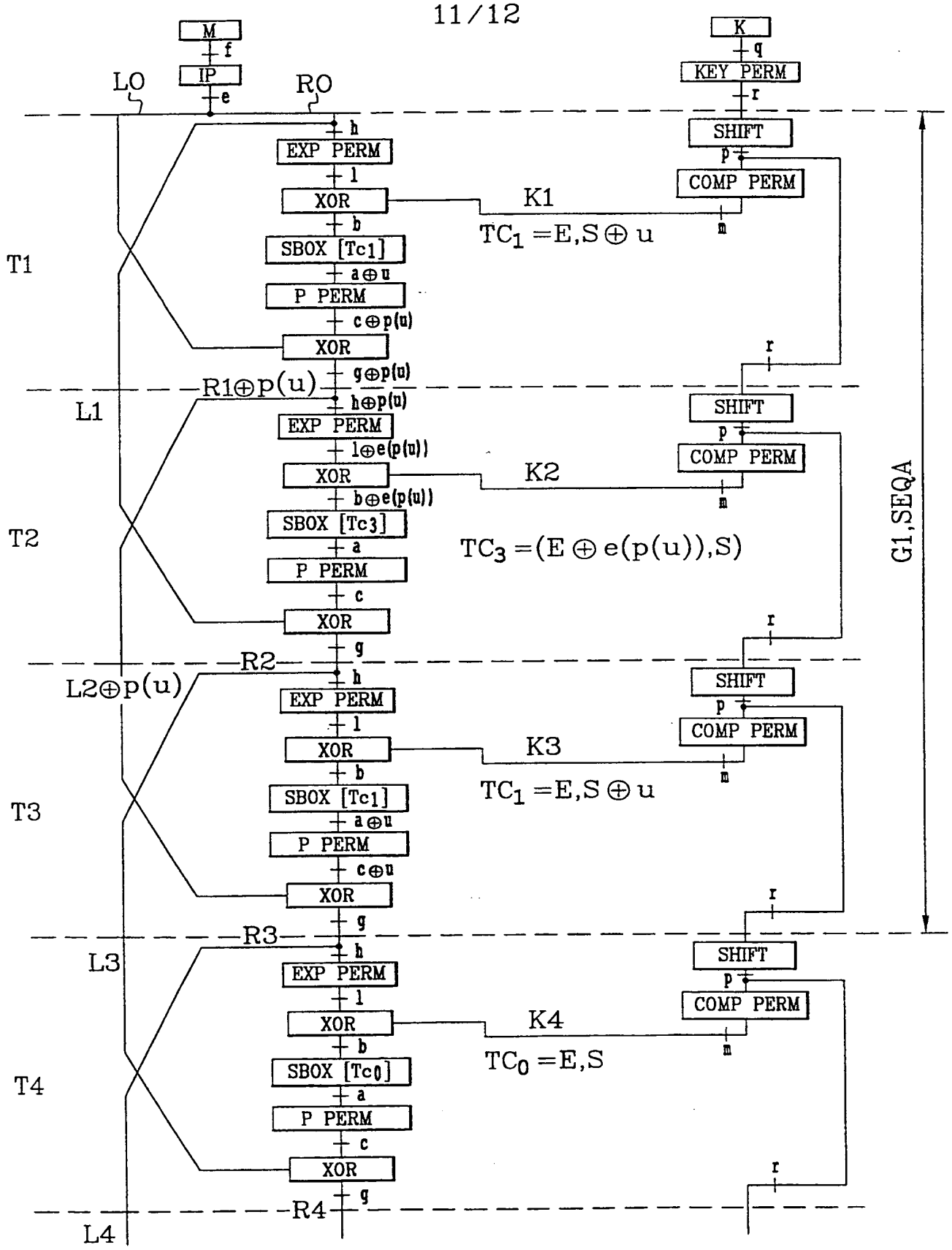

M
f
IP
e
L0
R0
K
q
KEY PERM
r
h
EXP PERM
l
XOR
K1
b
SBOX [Tc1]
TC1 =E,S⊕u
a⊕u
P PERM
c⊕p(u)
XOR
g⊕p(u)
SHIFT
p
COMP PERM
m
T1
R1⊕p(u)
L1
h⊕p(u)
EXP PERM
l⊕e(p(u))
XOR
K2
b⊕e(p(u))
SBOX [Tc3]
TC3 =(E⊕e(p(u)),S)
a
P PERM
c
XOR
g
T2
R2
L2⊕p(u)
h
EXP PERM
l
XOR
K3
b
SBOX [Tc1]
TC1 =E,S⊕u
a⊕u
P PERM
c⊕u
XOR
g
T3
R3
L3
h
EXP PERM
l
XOR
K4
b
SBOX [Tc0]
TC0 =E,S
a
P PERM
c
XOR
g
T4
R4
L4
G1,SEQA

FIG.12

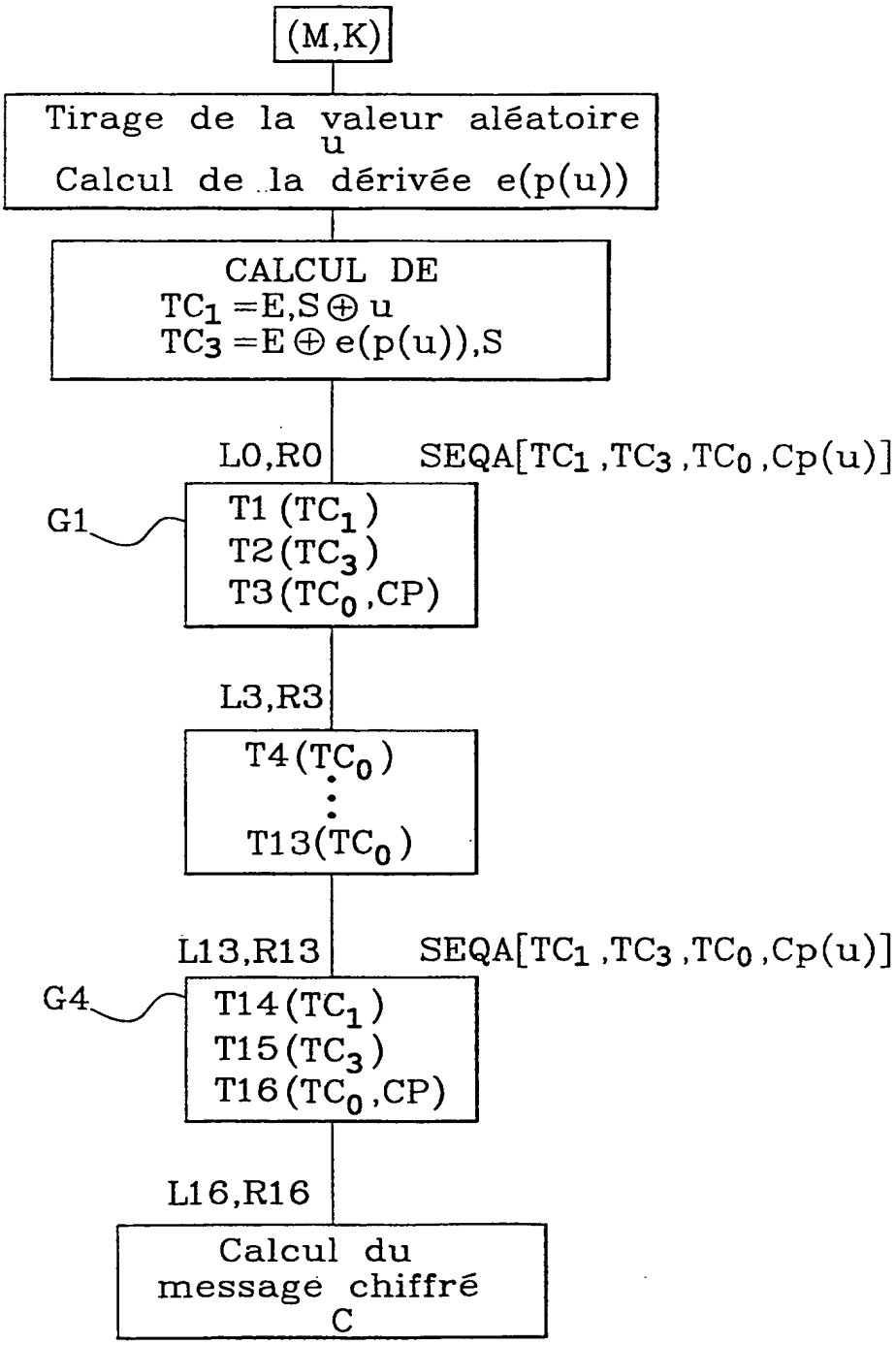

(M,K)
Tirage de la valeur aléatoire u
Calcul de la dérivée e(p(u))
CALCUL DE
TC1 =E,S ⊕ u
TC3 =E ⊕ e(p(u)),S
L0,R0
SEQA[TC1 ,TC3 ,TC0 ,Cp(u)]
G1
T1 (TC1)
T2(TC3)
T3(TC0 ,CP)
L3,R3
T4(TC0)
T13(TC0)
L13,R13
SEQA[TC1 ,TC3 ,TC0 ,Cp(u)]
G4
T14(TC1)
T15(TC3)
T16(TC0 ,CP)
L16,R16
Calcul du message chiffré C

FIG.13

INTERNATIONAL SEARCH REPORT

Inte .ional Application No
PCT/FR 99/02660

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 H04L9/06

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 H04L G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and. where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	MIYAGUCHI S: "SECRET KEY CIPHERS THAT CHANGE THE ENCIPHERMENT ALGORITHM UNDER THECONTROL OF THE KEY" NTT REVIEW, vol. 6, no. 4, 1 July 1994 (1994-07-01), pages 85-90, XP000460342 the whole document --- -/--	1,2,6,7

[X] Further documents are listed in the continuation of box C.

[X] Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
17 January 2000	26/01/2000

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL – 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Gautier, L

Form PCT/ISA/210 (second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

Int(ional Application No
PCT/FR 99/02660

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	YI X ET AL: "A METHOD FOR OBTAINING CRYPTOGRAPHICALLY STRONG 8X8 S-BOXES" IEEE GLOBAL TELECOMMUNICATIONS CONFERENCE, PHOENIX, ARIZONA, NOV. 3 - 8, 1997, vol. 2, 3 November 1997 (1997-11-03), pages 689-693, XP000737626 INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS abstract column 1, line 13 - line 29 column 2, line 6 - line 18 column 3, line 1 -column 5, line 1 ---	1-8
A	FR 2 672 402 A (GEMPLUS CARD INT) 7 August 1992 (1992-08-07) abstract page 1, line 4 - line 12 page 3, line 19 - line 23 figure 1 claim 1 -----	9,10

Form PCT/ISA/210 (continuation of second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

Information on patent family members

Int[illegible]ional Application No
PCT/FR 99/02660

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
FR 2672402 A	07-08-1992	NONE	

RAPPORT DE RECHERCHE INTERNATIONALE

De[illegible]de internationale No
PCT/FR 99/02660

A. CLASSEMENT DE L'OBJET DE LA DEMANDE
CIB 7 H04L9/06

Selon la classification internationale des brevets (CIB) ou à la fois selon la classification nationale et la CIB

B. DOMAINES SUR LESQUELS LA RECHERCHE A PORTE

Documentation minimale consultée (système de classification suivi des symboles de classement)
CIB 7 H04L G06F

Documentation consultée autre que la documentation minimale dans la mesure où ces documents relèvent des domaines sur lesquels a porté la recherche

Base de données électronique consultée au cours de la recherche internationale (nom de la base de données, et si réalisable, termes de recherche utilisés)

C. DOCUMENTS CONSIDERES COMME PERTINENTS

Catégorie °	Identification des documents cités, avec, le cas échéant, l'indication des passages pertinents	no. des revendications visées
A	MIYAGUCHI S: "SECRET KEY CIPHERS THAT CHANGE THE ENCIPHERMENT ALGORITHM UNDER THECONTROL OF THE KEY" NTT REVIEW, vol. 6, no. 4, 1 juillet 1994 (1994-07-01), pages 85-90, XP000460342 le document en entier --- -/--	1,2,6,7

[X] Voir la suite du cadre C pour la fin de la liste des documents

[X] Les documents de familles de brevets sont indiqués en annexe

° Catégories spéciales de documents cités:

"A" document définissant l'état général de la technique, non considéré comme particulièrement pertinent

"E" document antérieur, mais publié à la date de dépôt international ou après cette date

"L" document pouvant jeter un doute sur une revendication de priorité ou cité pour déterminer la date de publication d'une autre citation ou pour une raison spéciale (telle qu'indiquée)

"O" document se référant à une divulgation orale, à un usage, à une exposition ou tous autres moyens

"P" document publié avant la date de dépôt international, mais postérieurement à la date de priorité revendiquée

"T" document ultérieur publié après la date de dépôt international ou la date de priorité et n'appartenenant pas à l'état de la technique pertinent, mais cité pour comprendre le principe ou la théorie constituant la base de l'invention

"X" document particulièrement pertinent; l'inven tion revendiquée ne peut être considérée comme nouvelle ou comme impliquant une activité inventive par rapport au document considéré isolément

"Y" document particulièrement pertinent; l'inven tion revendiquée ne peut être considérée comme impliquant une activité inventive lorsque le document est associé à un ou plusieurs autres documents de même nature, cette combinaison étant évidente pour une personne du métier

"&" document qui fait partie de la même famille de brevets

Date à laquelle la recherche internationale a été effectivement achevée	Date d'expédition du présent rapport de recherche internationale
17 janvier 2000	26/01/2000

Nom et adresse postale de l'administration chargée de la recherche internationale	Fonctionnaire autorisé
Office Européen des Brevets, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Gautier, L

1

Formulaire PCT/ISA/210 (deuxième feuille) (juillet 1992)

RAPPORT DE RECHERCHE INTERNATIONALE

De. de Internationale No
PCT/FR 99/02660

C.(suite) DOCUMENTS CONSIDERES COMME PERTINENTS

Catégorie	Identification des documents cités. avec.le cas échéant. l'indicationdes passages pertinents	no. des revendications visees
A	YI X ET AL: "A METHOD FOR OBTAINING CRYPTOGRAPHICALLY STRONG 8X8 S-BOXES" IEEE GLOBAL TELECOMMUNICATIONS CONFERENCE, PHOENIX, ARIZONA, NOV. 3 - 8, 1997, vol. 2, 3 novembre 1997 (1997-11-03), pages 689-693, XP000737626 INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS abrégé colonne 1, ligne 13 - ligne 29 colonne 2, ligne 6 - ligne 18 colonne 3, ligne 1 -colonne 5, ligne 1 ---	1-8
A	FR 2 672 402 A (GEMPLUS CARD INT) 7 août 1992 (1992-08-07) abrégé page 1, ligne 4 - ligne 12 page 3, ligne 19 - ligne 23 figure 1 revendication 1 -----	9,10

Formulaire PCT/ISA/210 (suite de la deuxième feuille) (juillet 1992)

RAPPORT DE RECHERCHE INTERNATIONALE

Renseignements relatifs aux membres de familles de brevets

Den .e Internationale No
PCT/FR 99/02660

Document brevet cité au rapport de recherche	Date de publication	Membre(s) de la famille de brevet(s)	Date de publication
FR 2672402 A	07-08-1992	AUCUN	